Fig. 1

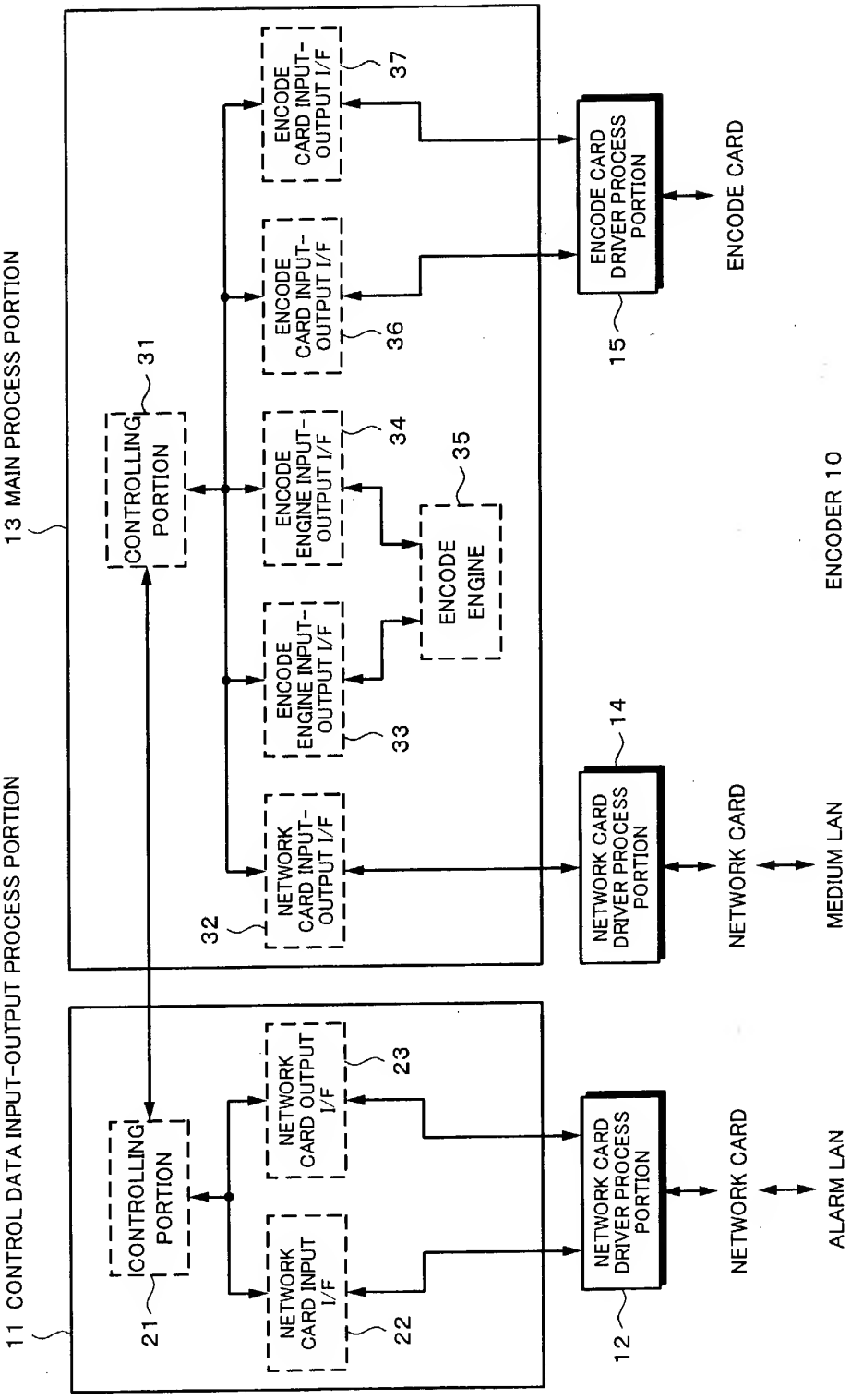
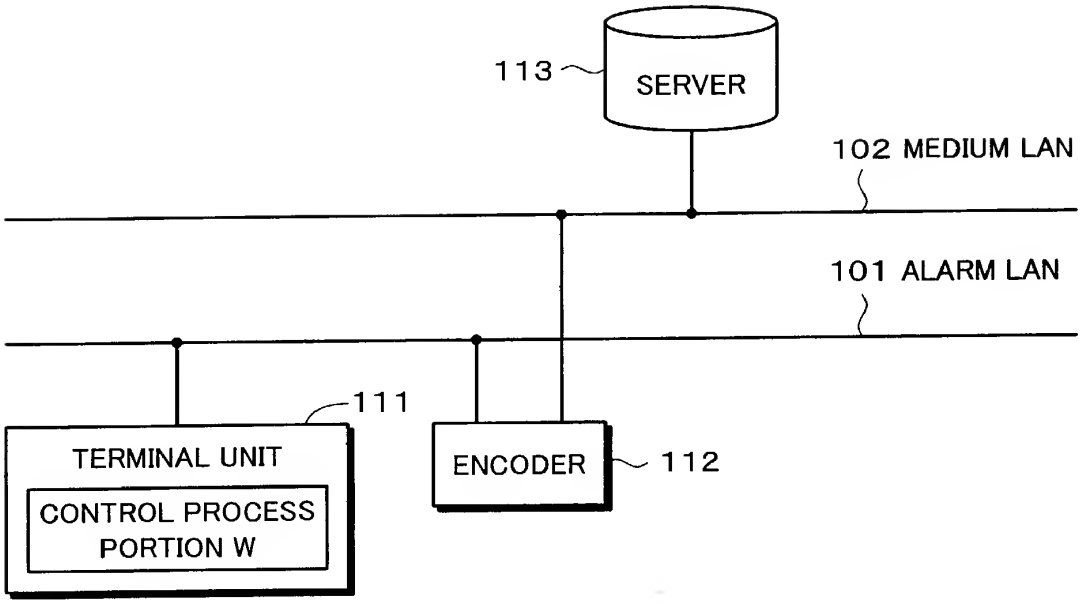

ENCODER 10
13 MAIN PROCESS PORTION
11 CONTROL DATA INPUT-OUTPUT PROCESS PORTION
CONTROLLING PORTION
31
ENCODE ENGINE
35
ENCODE ENGINE INPUT-OUTPUT I/F
34
33
ENCODE CARD INPUT-OUTPUT I/F
37
36
ENCODE CARD DRIVER PROCESS PORTION
15
ENCODE CARD
NETWORK CARD INPUT-OUTPUT I/F
32
NETWORK CARD DRIVER PROCESS PORTION
14
NETWORK CARD
MEDIUM LAN
CONTROLLING PORTION
21
NETWORK CARD OUTPUT I/F
23
NETWORK CARD INPUT I/F
22
NETWORK CARD DRIVER PROCESS PORTION
12
NETWORK CARD
ALARM LAN

Fig. 2



MUSIC DELIVERY SERVICE SYSTEM

Fig. 3

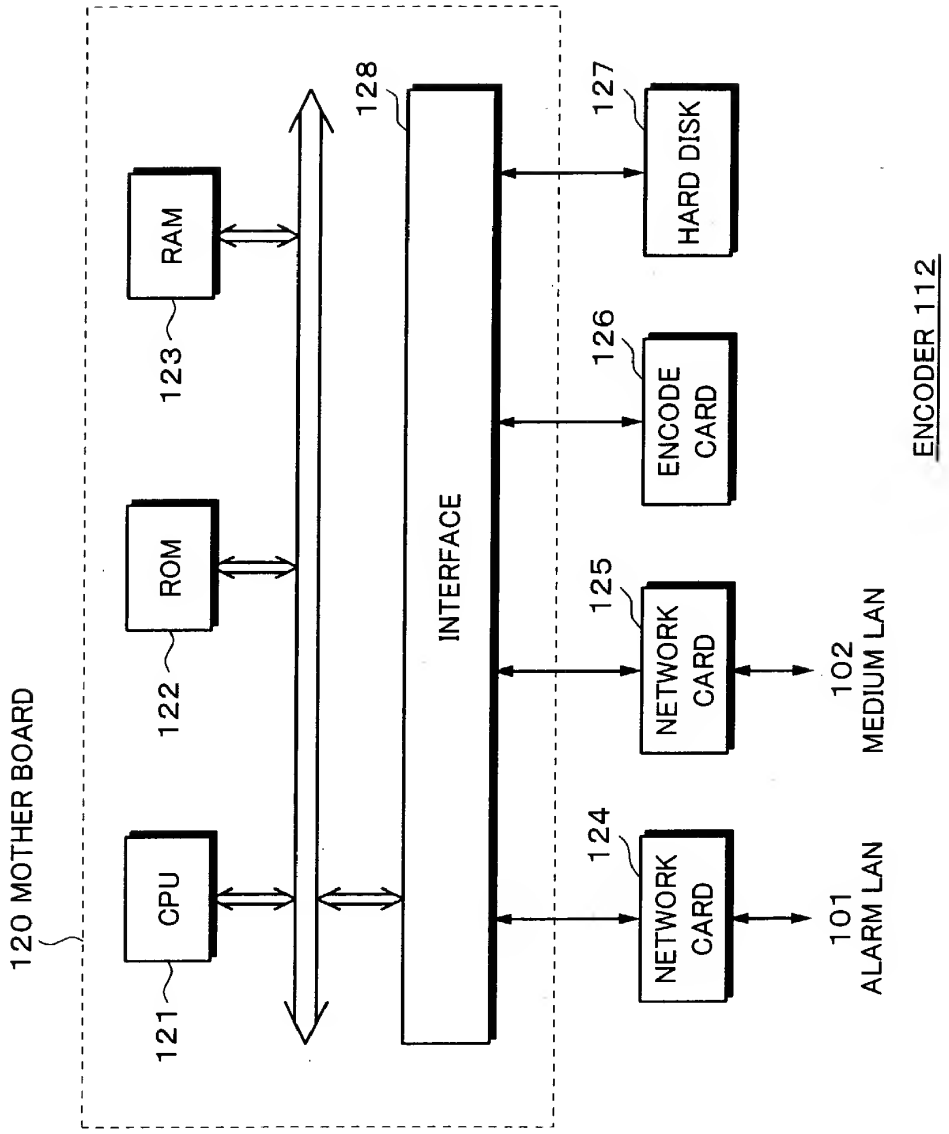

120 MOTHER BOARD
CPU
121
ROM
122
RAM
123
INTERFACE
128
NETWORK CARD
124
NETWORK CARD
125
ENCODE CARD
126
HARD DISK
127
101 ALARM LAN
102 MEDIUM LAN
ENCODER 112

Fig. 4

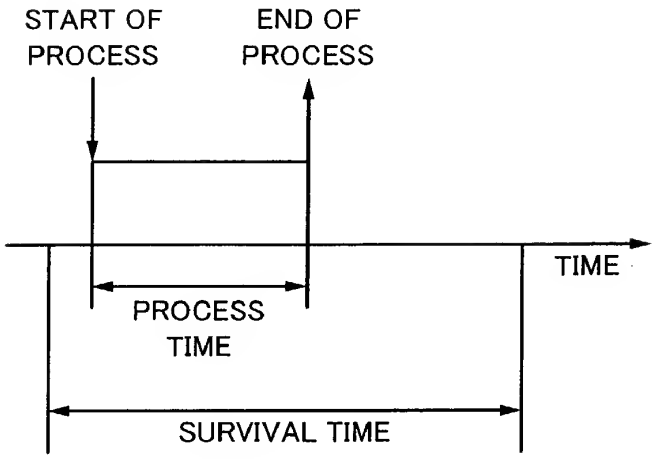

START OF PROCESS
END OF PROCESS
TIME
PROCESS TIME
SURVIVAL TIME

Fig. 5

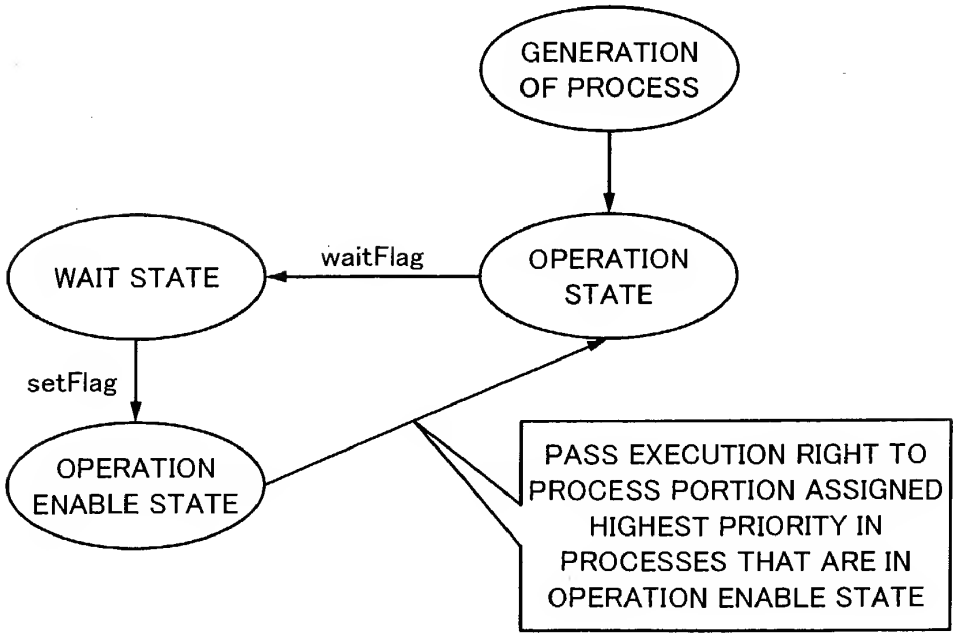

GENERATION OF PROCESS
WAIT STATE
waitFlag
OPERATION STATE
setFlag
OPERATION ENABLE STATE
PASS EXECUTION RIGHT TO PROCESS PORTION ASSIGNED HIGHEST PRIORITY IN PROCESSES THAT ARE IN OPERATION ENABLE STATE

Fig. 6

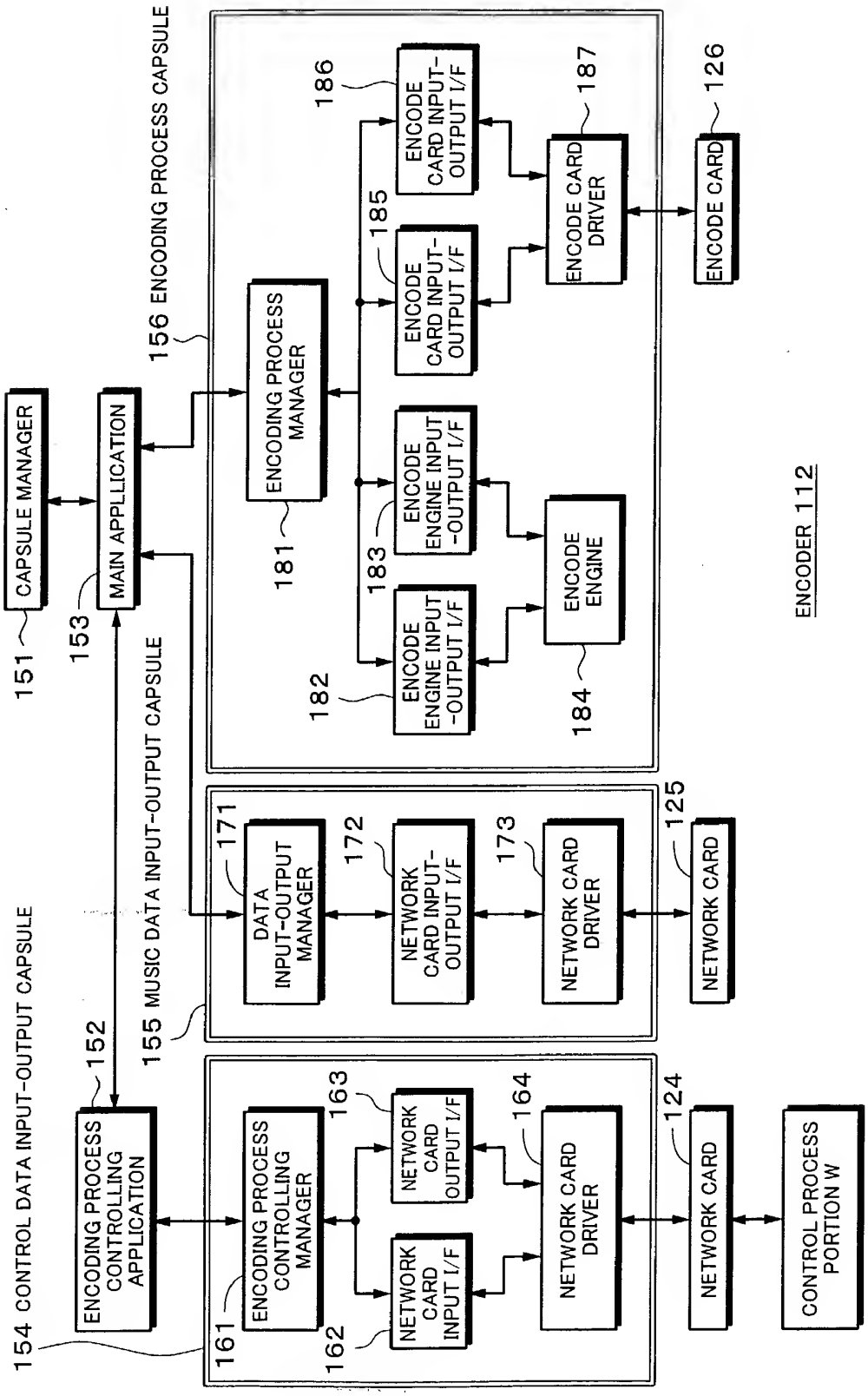

CAPSULE MANAGER
151
MAIN APPLLICATION
153
156 ENCODING PROCESS CAPSULE
ENCODING PROCESS MANAGER
181
ENCODE ENGINE INPUT -OUTPUT I/F
182
ENCODE ENGINE
184
ENCODE ENGINE INPUT -OUTPUT I/F
183
ENCODE CARD INPUT-OUTPUT I/F
185
ENCODE CARD INPUT-OUTPUT I/F
186
ENCODE CARD DRIVER
187
ENCODE CARD
126
154 CONTROL DATA INPUT-OUTPUT CAPSULE
152
155 MUSIC DATA INPUT-OUTPUT CAPSULE
DATA INPUT-OUTPUT MANAGER
171
NETWORK CARD INPUT-OUTPUT I/F
172
NETWORK CARD DRIVER
173
NETWORK CARD
125
ENCODING PROCESS CONTROLLING APPLICATION
ENCODING PROCESS CONTROLLING MANAGER
161
NETWORK CARD INPUT I/F
162
NETWORK CARD OUTPUT I/F
163
NETWORK CARD DRIVER
164
NETWORK CARD
124
CONTROL PROCESS PORTION W
ENCODER 112

Fig. 7
Fig. 7A
Fig. 7B

Fig. 7A

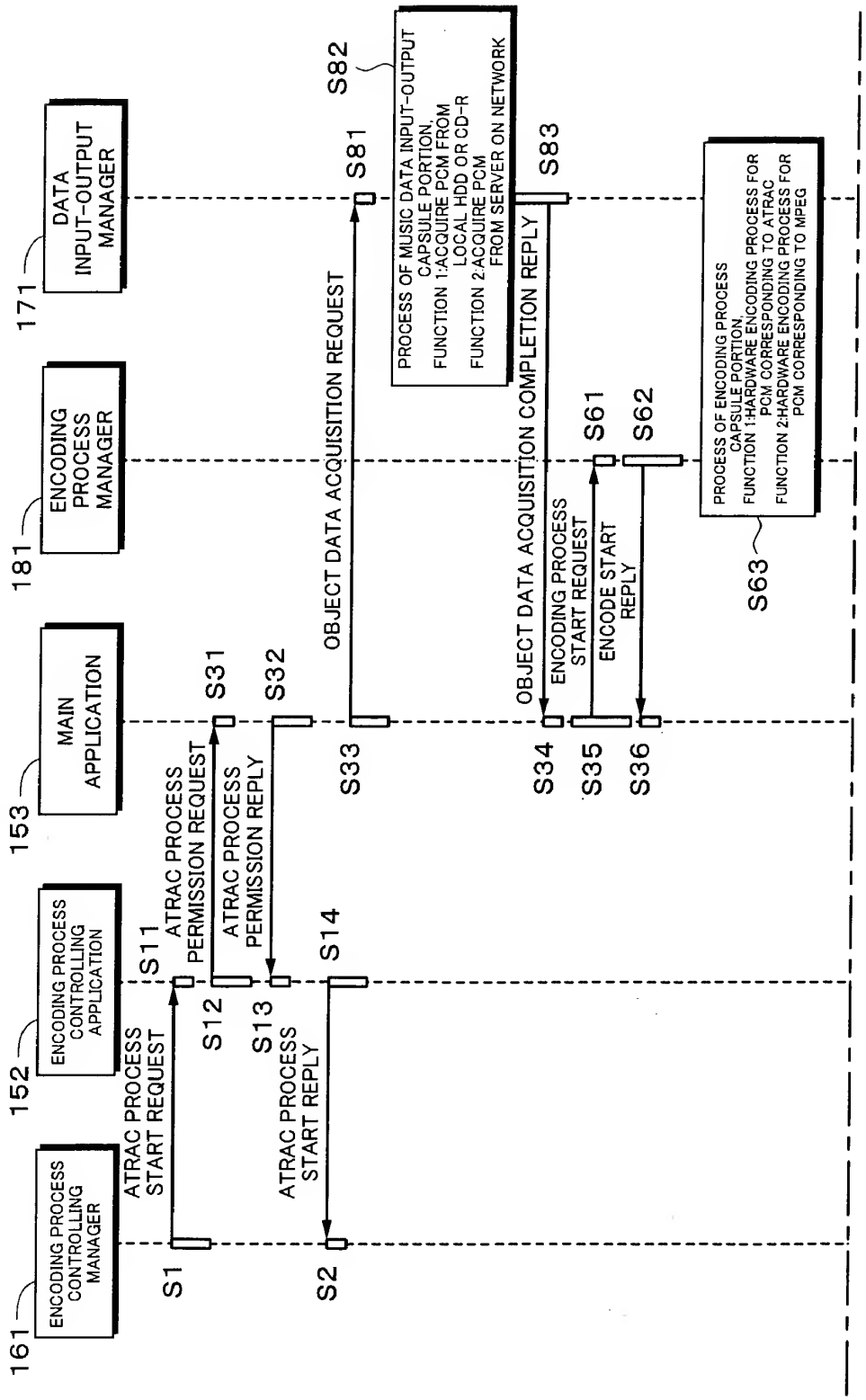


Fig. 7B

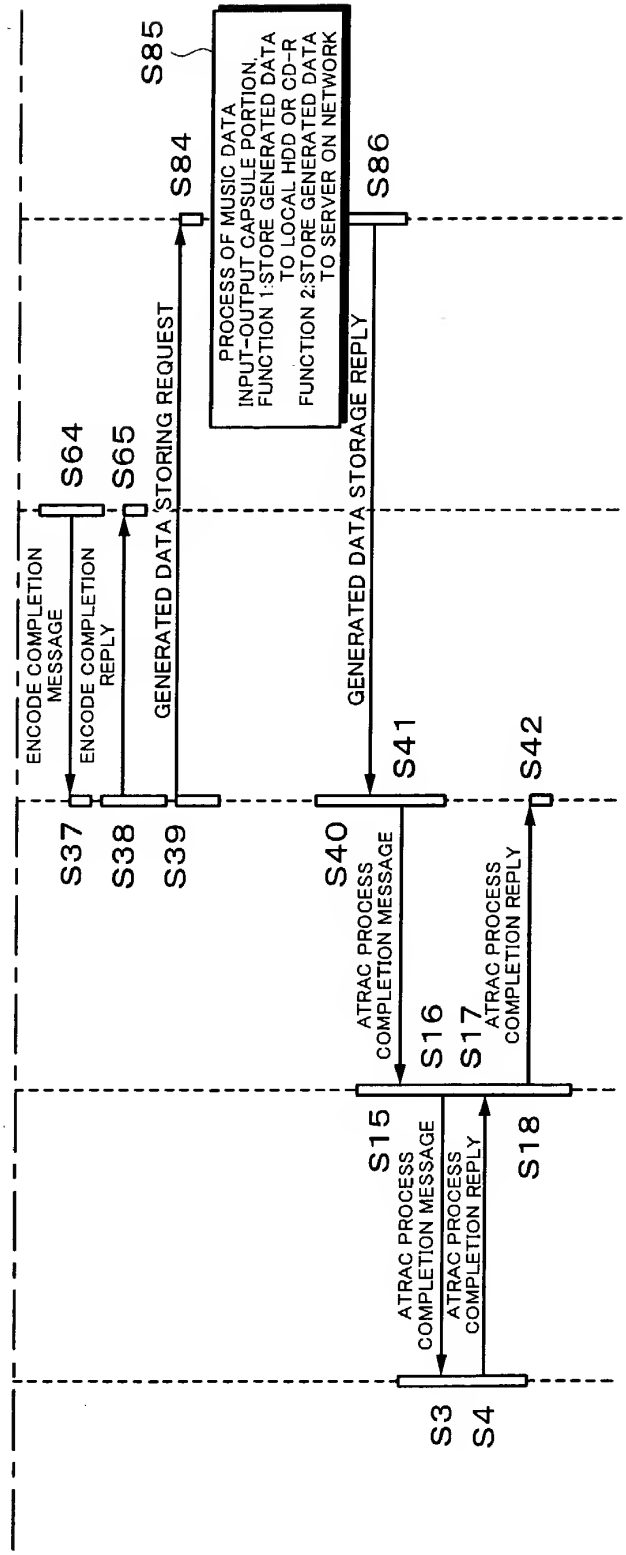
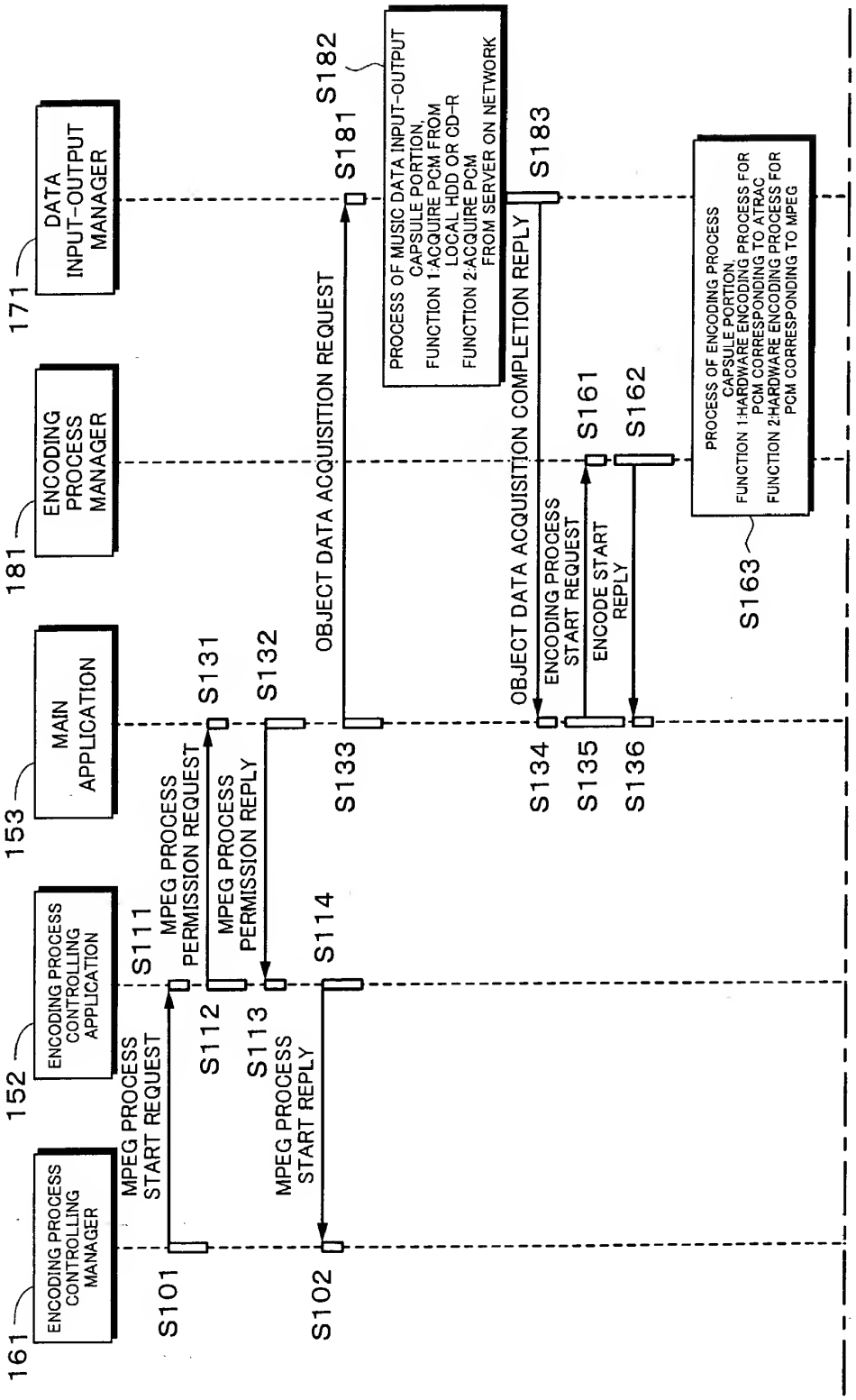

S3
S4
ATRAC PROCESS COMPLETION MESSAGE
ATRAC PROCESS COMPLETION REPLY
S15
S16
S17
S18
ATRAC PROCESS COMPLETION MESSAGE
ATRAC PROCESS COMPLETION REPLY
S37
S38
S39
S40
S41
S42
ENCODE COMPLETION MESSAGE
ENCODE COMPLETION REPLY
GENERATED DATA STORING REQUEST
GENERATED DATA STORAGE REPLY
S64
S65
S84
S85
S86
PROCESS OF MUSIC DATA
INPUT-OUTPUT CAPSULE PORTION.
FUNCTION 1:STORE GENERATED DATA TO LOCAL HDD OR CD-R
FUNCTION 2:STORE GENERATED DATA TO SERVER ON NETWORK

Fig. 8
Fig. 8A
Fig. 8B

Fig. 8A



09869658 "052001 15:36:39

Fig. 8B

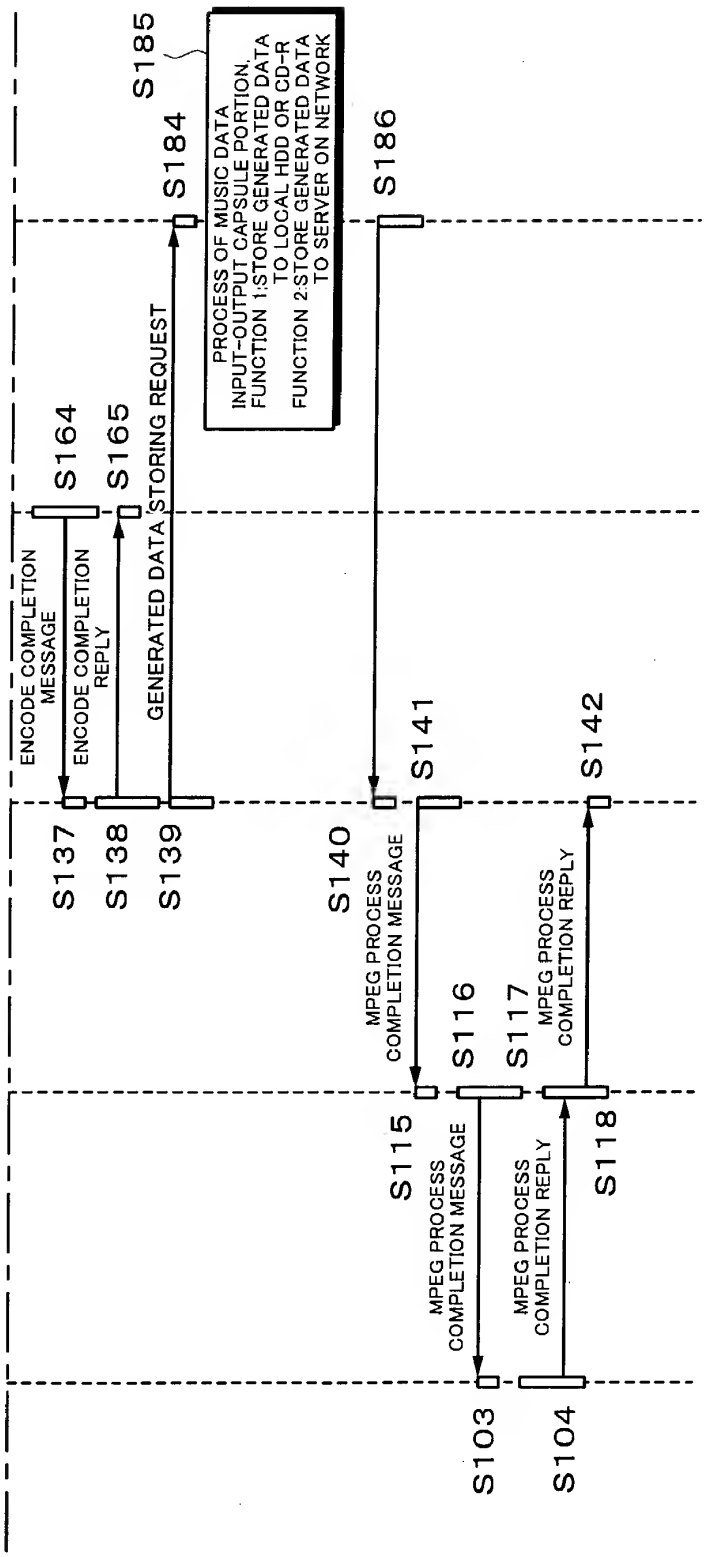


Fig. 9

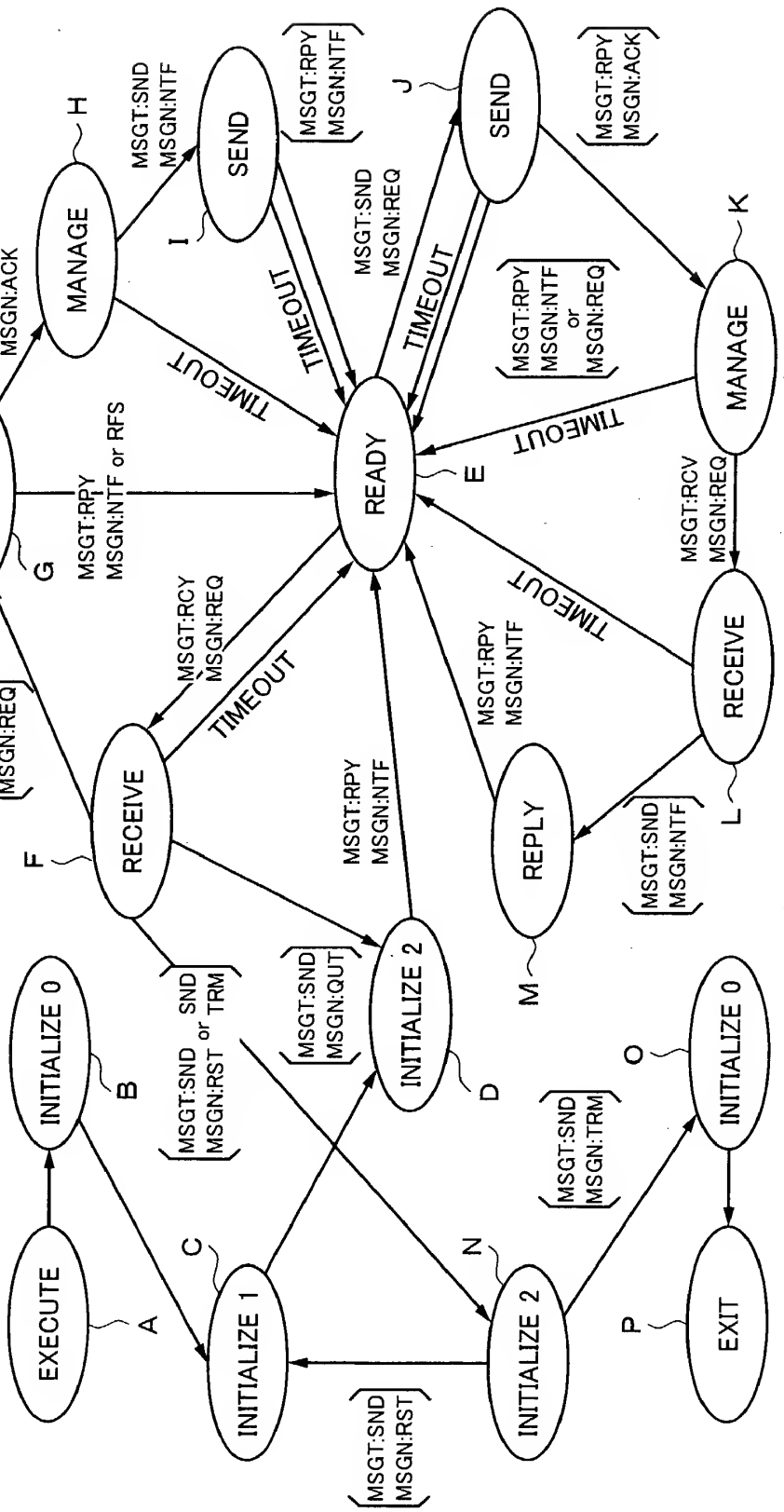

EXECUTE
A
INITIALIZE 0
B
INITIALIZE 1
C
INITIALIZE 2
D
RECEIVE
F
REPLY
G
MANAGE
H
SEND
I
READY
E
SEND
J
MANAGE
K
RECEIVE
L
REPLY
M
INITIALIZE 2
N
INITIALIZE 0
O
EXIT
P
MSGT:SND
MSGN:RST
MSGT:SND
MSGN:REQ
MSGT:SND
MSGN:RST
or
SND
TRM
MSGT:SND
MSGN:QUT
MSGT:RPY
MSGN:NTF
MSGT:RCY
MSGN:REQ
MSGT:RPY
MSGN:ACK
MSGT:RPY
MSGN:NTF
or RFS
MSGT:SND
MSGN:NTF
MSGT:RPY
MSGN:NTF
MSGT:SND
MSGN:REQ
MSGT:RPY
MSGN:NTF
MSGT:RPY
MSGN:ACK
MSGT:RPY
MSGN:NTF
or
MSGN:REQ
MSGT:RCV
MSGN:REQ
MSGT:SND
MSGN:NTF
MSGT:SND
MSGN:TRM
TIMEOUT
TIMEOUT
TIMEOUT
TIMEOUT
TIMEOUT

Fig. 10

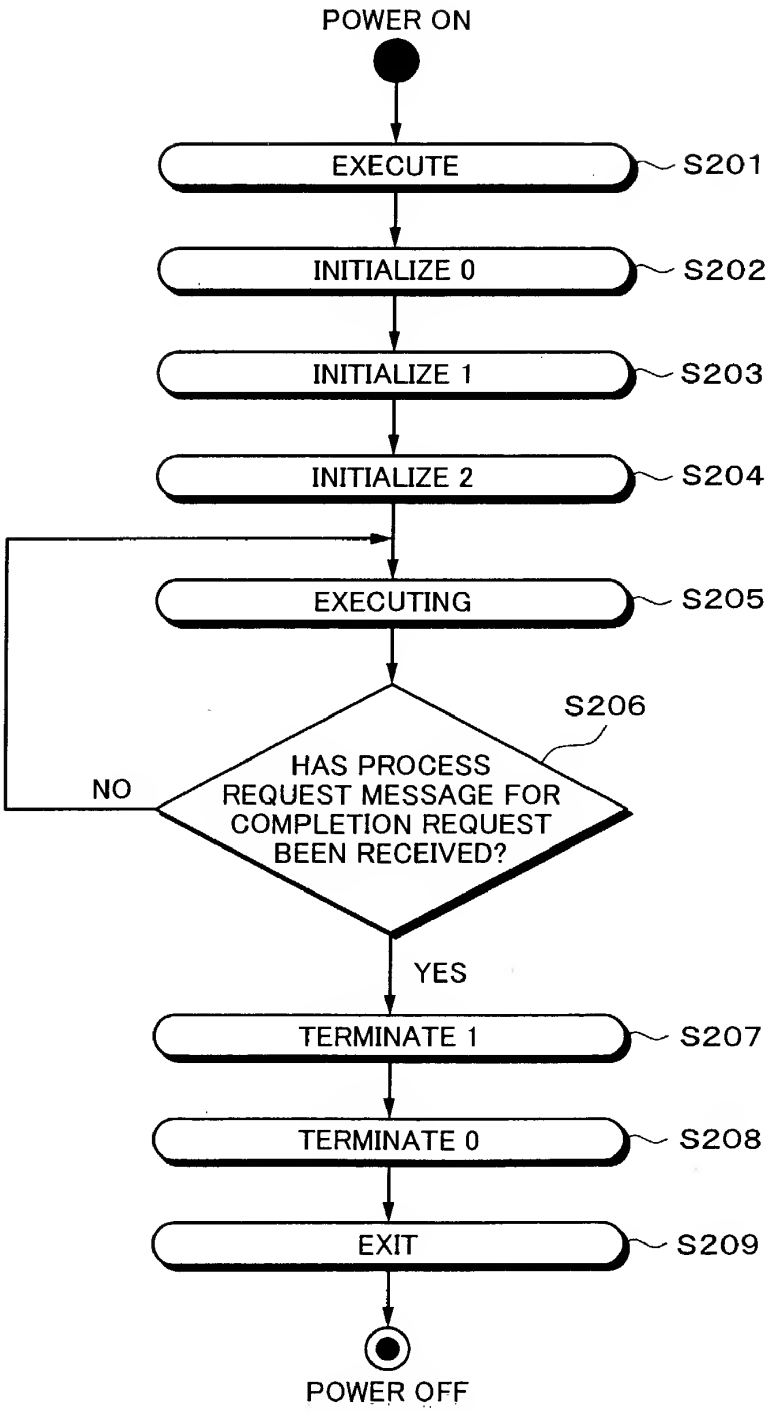

POWER ON
EXECUTE
S201
INITIALIZE 0
S202
INITIALIZE 1
S203
INITIALIZE 2
S204
EXECUTING
S205
S206
HAS PROCESS REQUEST MESSAGE FOR COMPLETION REQUEST BEEN RECEIVED?
NO
YES
TERMINATE 1
S207
TERMINATE 0
S208
EXIT
S209
POWER OFF

Fig. 11A

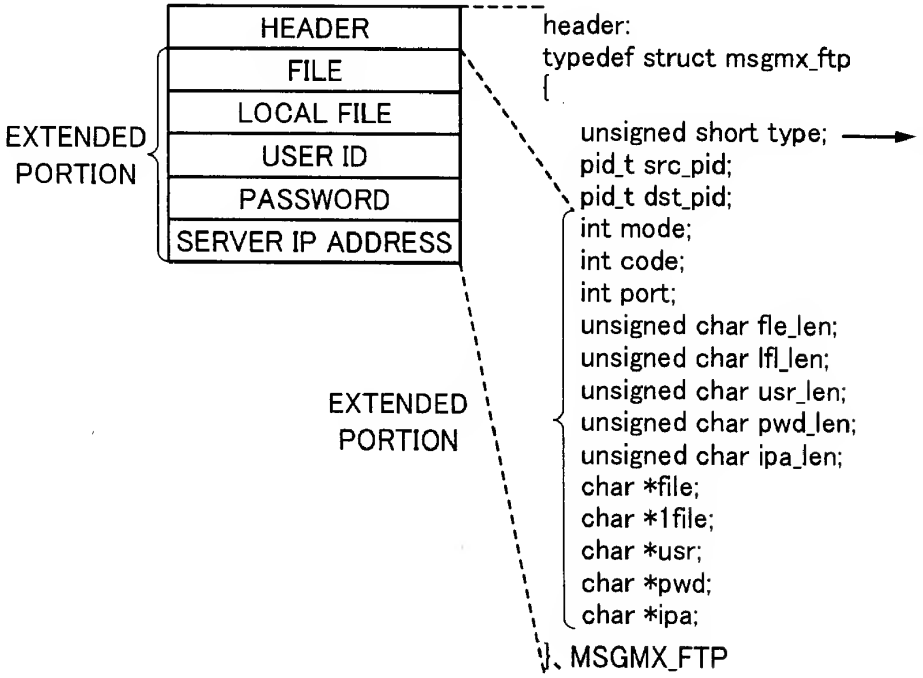
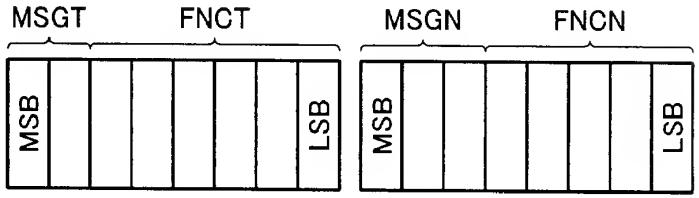


Fig. 11B

```
header:
typedef struct msgmx_ftp
{
    unsigned short type;
    pid_t src_pid;
    pid_t dst_pid;
    int mode;
    int code;
    int port;
    unsigned char fle_len;
    unsigned char lfl_len;
    unsigned char usr_len;
    unsigned char pwd_len;
    unsigned char ipa_len;
    char *file;
    char *1file;
    char *usr;
    char *pwd;
    char *ipa;
} MSGMX_FTP
```

Fig. 11C



FUJESU-P3309603

09/869658

Fig. 12A

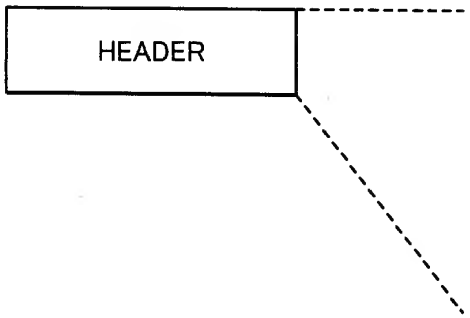


Fig. 12B

```
header_rpy:
typedef struct msgmx_ftp_rpy
{
    unsigned short structs:
    pid_t src_pid:
    pid_t dst_pid:
}, MSGMX_FTP_RPY
```

09869658 062501

Fig. 13

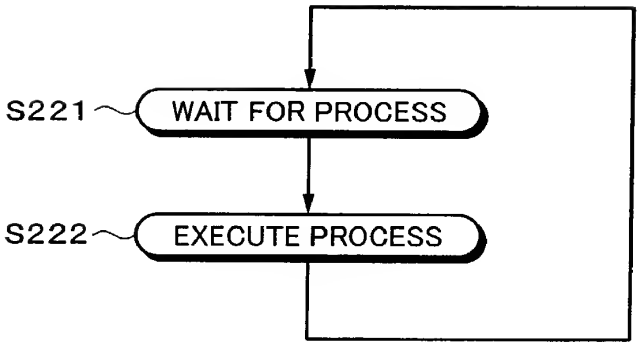


Fig. 14

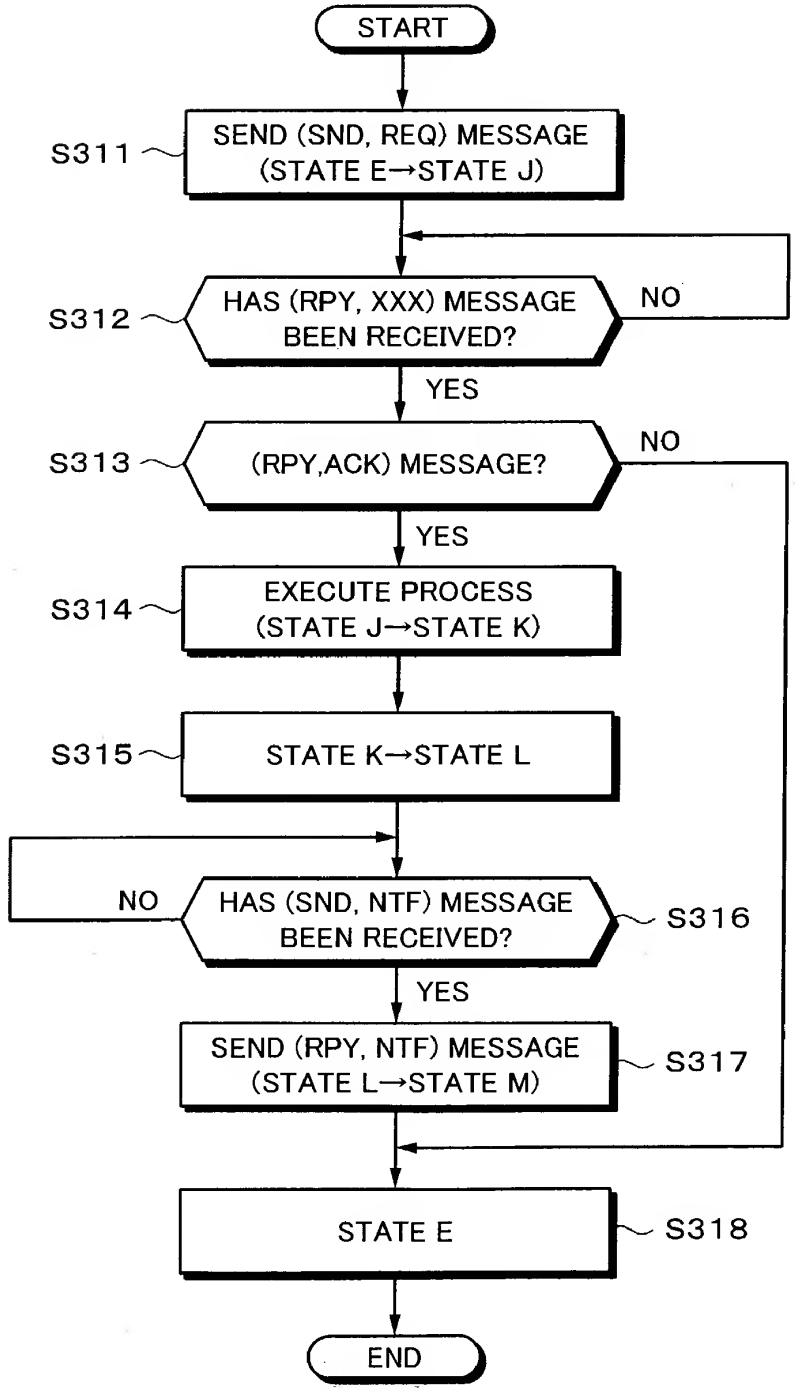

START
S311
SEND (SND, REQ) MESSAGE
(STATE E→STATE J)
S312
HAS (RPY, XXX) MESSAGE
BEEN RECEIVED?
NO
YES
S313
(RPY,ACK) MESSAGE?
NO
YES
S314
EXECUTE PROCESS
(STATE J→STATE K)
S315
STATE K→STATE L
NO
HAS (SND, NTF) MESSAGE
BEEN RECEIVED?
S316
YES
SEND (RPY, NTF) MESSAGE
(STATE L→STATE M)
S317
STATE E
S318
END

Fig. 15

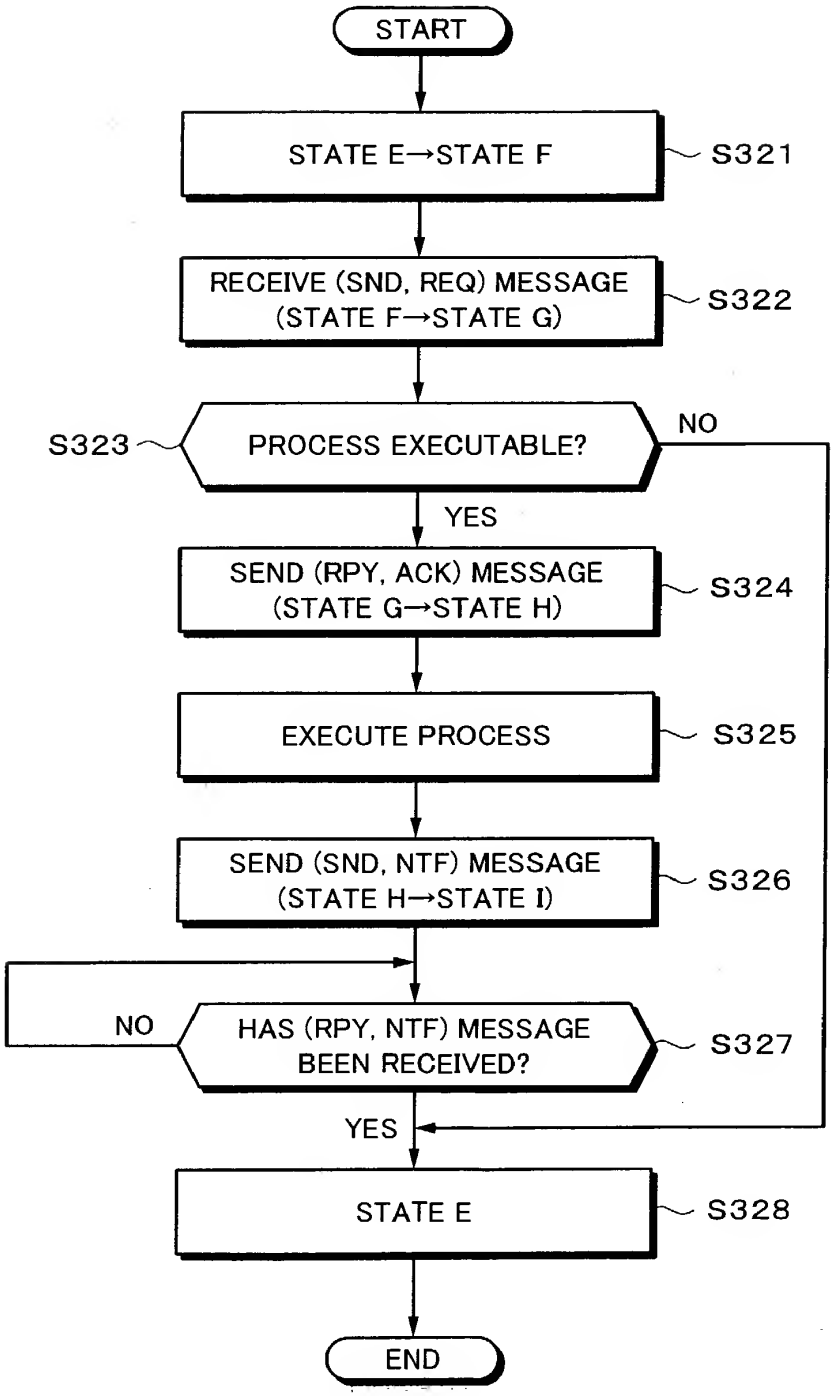

START
STATE E→STATE F
S321
RECEIVE (SND, REQ) MESSAGE
(STATE F→STATE G)
S322
S323
PROCESS EXECUTABLE?
NO
YES
SEND (RPY, ACK) MESSAGE
(STATE G→STATE H)
S324
EXECUTE PROCESS
S325
SEND (SND, NTF) MESSAGE
(STATE H→STATE I)
S326
NO
HAS (RPY, NTF) MESSAGE
BEEN RECEIVED?
S327
YES
STATE E
S328
END

Fig. 16

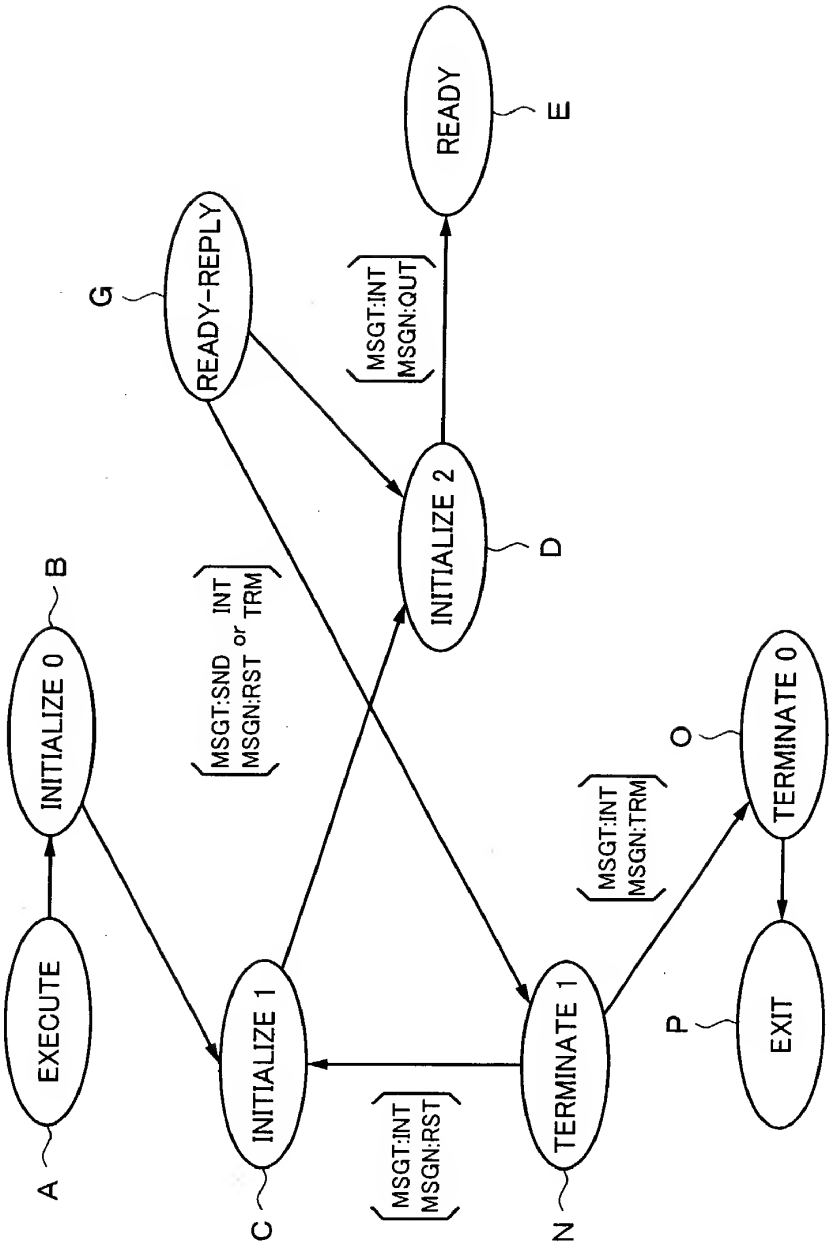

EXECUTE
A
INITIALIZE 0
B
INITIALIZE 1
C
READY-REPLY
G
INITIALIZE 2
D
READY
E
TERMINATE 1
N
TERMINATE 0
O
EXIT
P
MSGT:SND
MSGN:RST
INT
or
TRM
MSGT:INT
MSGN:QUT
MSGT:INT
MSGN:RST
MSGT:INT
MSGN:TRM

Fig. 17

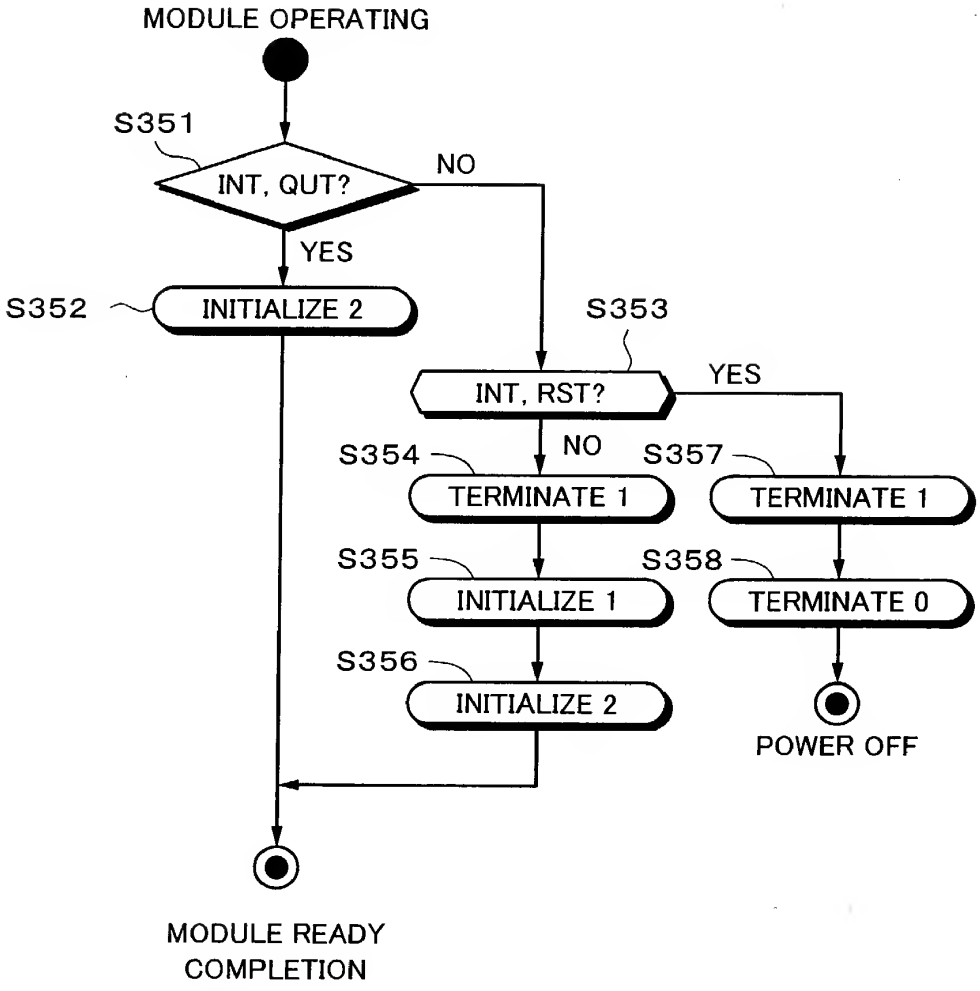

MODULE OPERATING
S351
INT, QUT?
NO
YES
S352
INITIALIZE 2
S353
INT, RST?
YES
NO
S354
TERMINATE 1
S357
TERMINATE 1
S355
INITIALIZE 1
S358
TERMINATE 0
S356
INITIALIZE 2
POWER OFF
MODULE READY
COMPLETION

Fig. 18

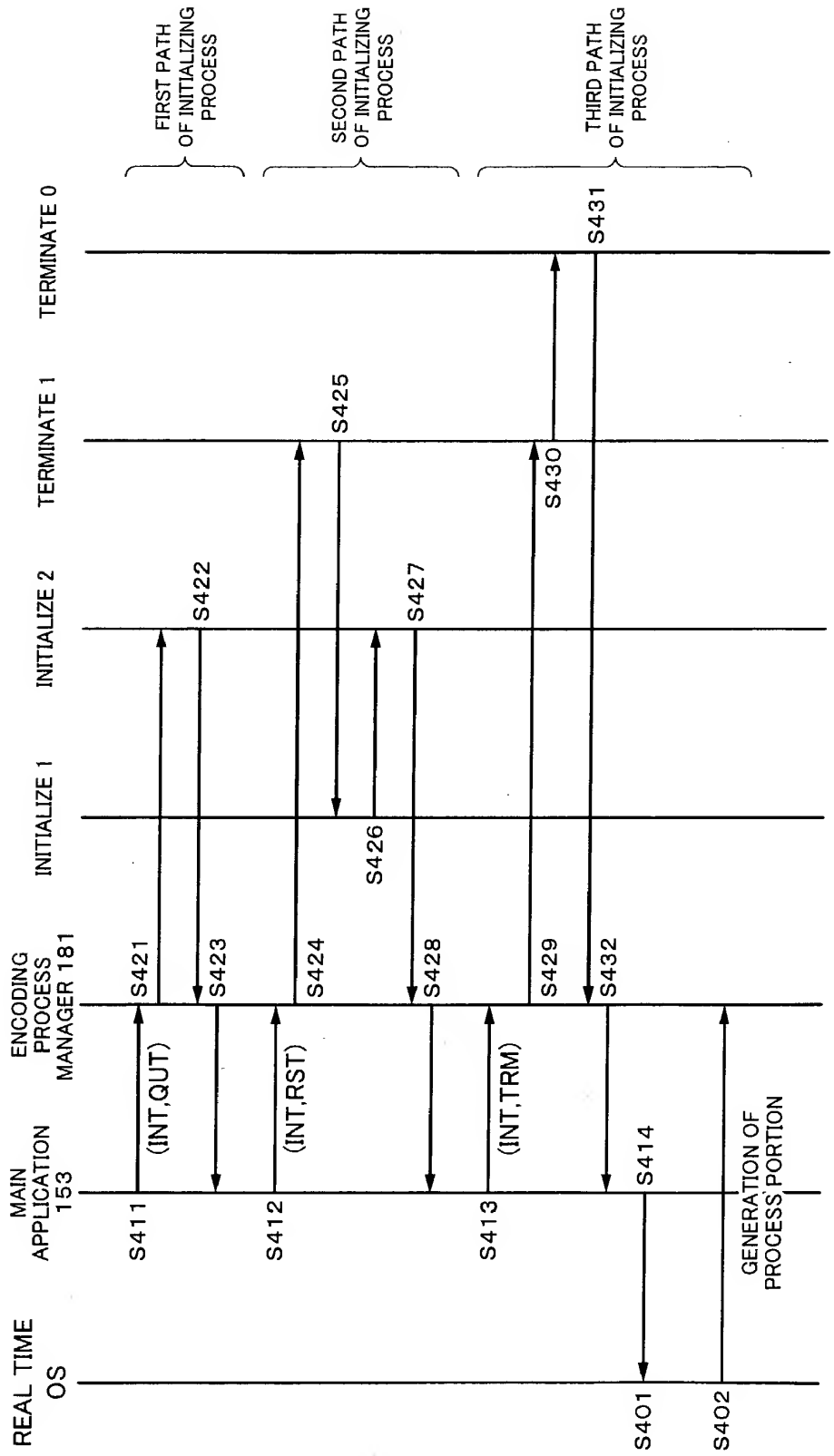
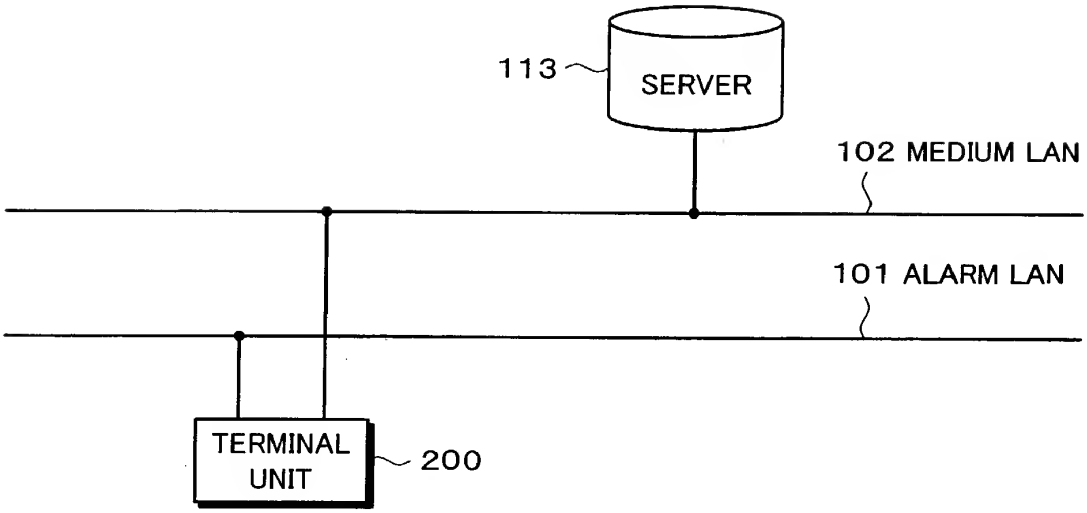

REAL TIME OS
MAIN APPLICATION 153
ENCODING PROCESS MANAGER 181
INITIALIZE 1
INITIALIZE 2
TERMINATE 1
TERMINATE 0
FIRST PATH OF INITIALIZING PROCESS
SECOND PATH OF INITIALIZING PROCESS
THIRD PATH OF INITIALIZING PROCESS
S411
S412
S413
S414
S421
S422
S423
S424
S425
S426
S427
S428
S429
S430
S431
S432
S401
S402
(INT,QUT)
(INT,RST)
(INT,TRM)
GENERATION OF PROCESS PORTION

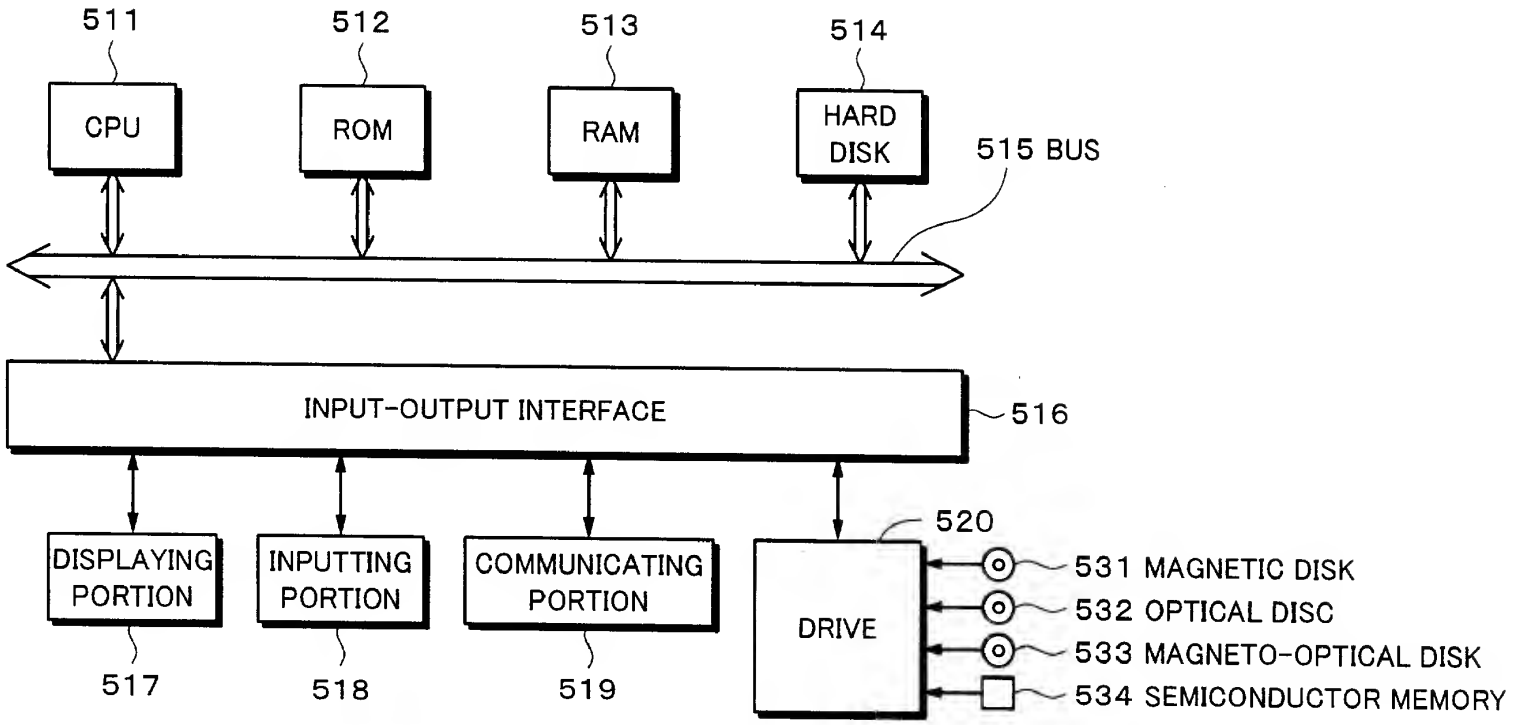
Fig. 19



MUSIC DELIVERY SERVICE SYSTEM

09/869658

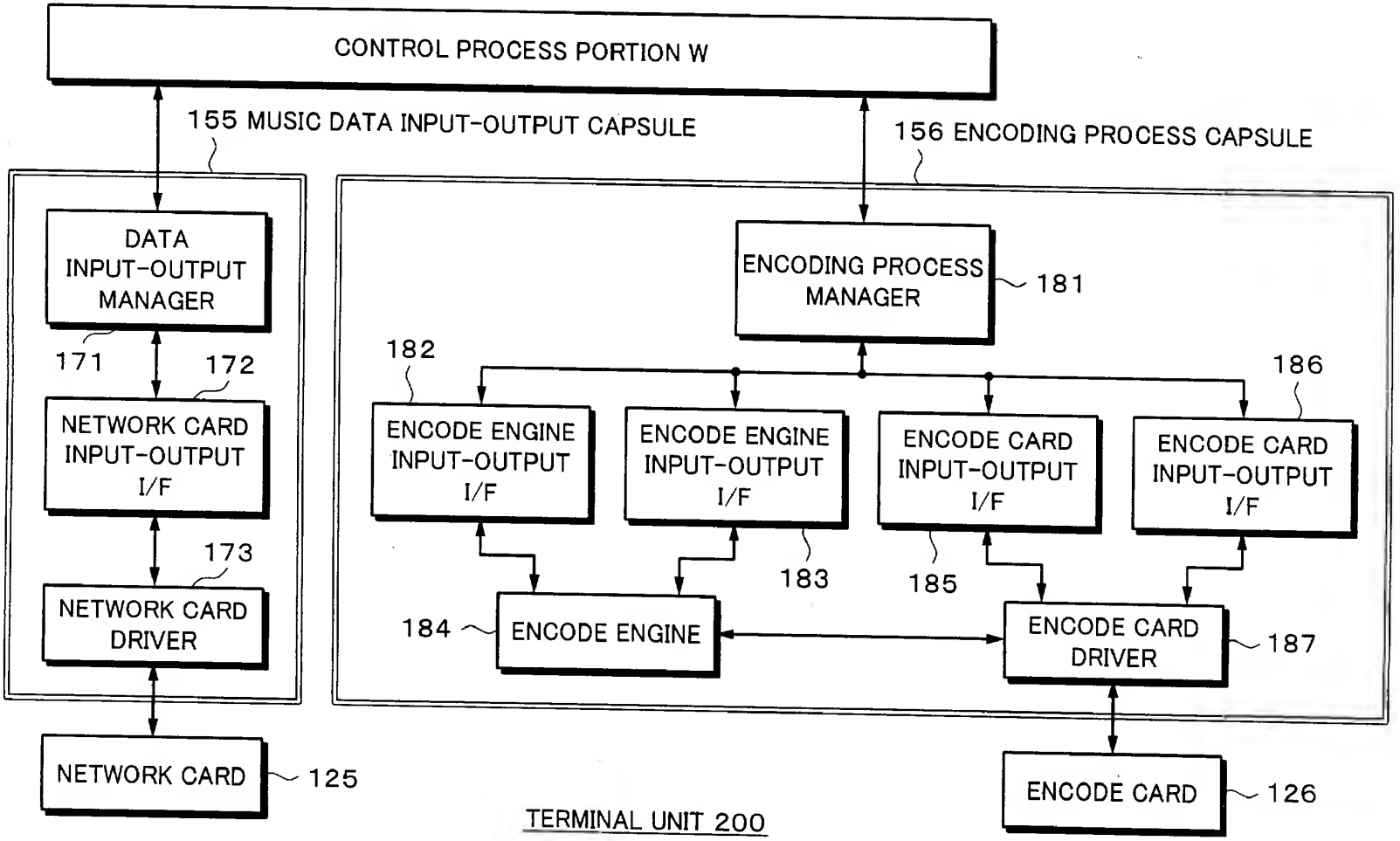
Fig. 20



TERMINAL UNIT 200

09/869658

Fig. 21



151 ... CAPSULE MANAGER

152 ... ENCODING PROCESS CONTROLLING APPLICATION

153 ... MAIN APPLICATION

154 ... CONTROL DATA INPUT - OUTPUT CAPSULE

155 ... MUSIC DATA INPUT - OUTPUT CAPSULE

156 ... ENCODING PROCESS CAPSULE

161 ... ENCODING PROCESS CONTROLLING MANGER

162 ... NETWORK CARD INPUT I/F PROCESS

163 ... NETWORK CARD OUTPUT I/F

164 ... NETWORK CARD DRIVER

171 ... DATA INPUT - OUTPUT MANAGER

172 ... NETWORK CARD INPUT - OUTPUT I/F

173 ... NETWORK CARD DRIVER

181 ... ENCODING PROCESS MANAGER

182 ... ENCODE ENGINE INPUT - OUTPUT I/F

183 ... ENCODE ENGINE INPUT - OUTPUT I/F

184 ... ENCODE ENGINE

185 ... ENCODE CARD INPUT - OUTPUT I/F

186 ... ENCODE CARD INPUT - OUTPUT I/F

187 ... ENCODE CARD DRIVER